目录

1　降维方法总结

1.1　多维缩放

若要求原始空间中样本之间的距离在低维空间中得以保持，即得到“多维缩放”（Multiple Dimensional Scaling，简称 MDS）这样一种经典算法。

MDS 巧妙的将低维样本的内积矩阵与原样本距离矩阵进行近似，通过特征值分解将距离相同这一目标在低维样本上得到一定的保留。

1.2　主成分分析

一个三维物体, 选取一个面进行投影, 得到的就是一个二维平面.PCA 算法就是一个投影算法, 将一个高维空间投影到一个低维空间当中.

PCA 算法的目标就是找到这样一个超平面, 能够让这个投影保留更大的特征, 对所有样本进行恰当的表达.

PCA 假设原空间存在一个线性变换 **W**, 从上面取得 d' 个最大特征值的特征向量方向, 让原空间的元素投影到这个 d' 空间, 并让元素之间距离差之和最小.

1.3　核主成分分析

Kernel PCA 是基于核技巧对线性降维方法 PCA 进行核化。使得线性不可分的高维样本通过非线性映射找到恰当的低维嵌入。

可以想象一个瑞士卷，不管选择什么样的方法进行“切割”都会失去其原有的结构。

KPCA 的思想就是，把原始的非线性的数据映射到高维空间变成线性的，然后用 PCA 来处理映射后的高维数据。

对于不显示的映射函数，引入了核函数 $K = \phi(X)\phi(X)^{\mathrm{T}}$，通过特征值分解，取 K 最大的 d 个的特征值对应的特征向量组成变换矩阵进行求解。

1.4 流形学习

特点是借助流形局部具有欧式空间的性质，能够用欧式距离进行计算。

1.4.1 等度量映射

Isomap 的基本出发点，是认为低维流形嵌入到高维空间后，直接在高维空间计算直线距离具有误导性，因为高维空间中的直线距离在低维流形上是不可达的。

因此 Isomap 将样本间距离计算从欧式距离改为"测地线"距离，其具体计算方式是对样本通过 k 近邻进行建图，通过 Dijkstra 或者 Floyd 算法计算两点距离，将距离矩阵 **D** 作为 MDS 算法的输入。

需要注意的是近邻图的构建有两种方法：

1. 设置近邻点个数。
2. 设置近邻阈值。

显然无论是哪种方法，都有可能因为不恰当的设置导致连通图断路或者将相距较远的样本连接起来。

1.4.2 局部线性嵌入

LLE 的思想就是，一个流形在很小的局部邻域上可以近似看成欧式的，就是局部线性的。那么，在小的局部邻域上，一个点就可以用它周围的点在最小二乘意义下最优的线性表示。LLE 把这个线性拟合的系数当成这个流形局部几何性质的刻画。

那么一个好的低维表示，就应该也具有同样的局部几何。

所以 LLE 的目标就是在高维空间中样本与其近邻点的重构关系在低维空间中得以保持。

1.5 度量学习

降维的核心在在于寻找合适空间，而合适空间的定义就是距离度量，而度量学习就是尝试学习出一种合适的距离度量。

对于欧式距离，若假定不同正交方向的重要性不同，可引入属性权重 w，对于不同的方向，若假定其之间并不是无关的，即认为某些属性之间是相关的，此时其对应的坐标轴不再正交。因此引入半正定对称矩阵 $\mathbf{M}$，于是得到了马氏距离：

$$dish^2_{mah}(x_i, x_j) = (x_i - x_j)^T \mathbf{M}(x_i - x_j) = ||x_i - x_j||^2_M$$

对于不同的学习目标可以学习到不同的距离度量矩阵。

2 MDS 算法基本原理和证明

输入:

距离矩阵 $\mathbf{D} \in \mathbb{R}^{m\times m}$, 其元素 $dist_{ij}$ 为样本 x_i 到 x_j 的距离;

低维空间维数 d';

过程:

1. 计算 $dist_{i\cdot}^2, dist_{\cdot i}^2, dist_{\cdot\cdot}^2$;

2. 计算矩阵 **B**;

3. 对矩阵 **B** 做特征值分解;

4. 取 $\tilde{\Lambda}$ 为 d' 个最大特征值所构成的对角矩阵,$\tilde{\mathbf{V}}$ 为相应的特征向量矩阵;

输出:

矩阵 $\tilde{\mathbf{V}}\tilde{\Lambda}^{1/2} \in \mathbb{R}^{m\times d'}$, 每行是一个样本的低维坐标;

假定 m 个样本在原始空间的距离矩阵为 $\mathbf{D} \in \mathbb{R}^{m\times m}$, 其第 i 行 j 列的元素 $dist_{ij}$ 为样本 x_i 到 x_j 的距离. 我们的目标是获得样本在 d' 维空间的坐标表示 $\mathbf{Z} \in \mathbb{R}^{d'\times m}, d' \leq d$. 且满足任意两样本距离不变, 即在 d' 维空间中的欧式距离等于原始空间中的距离——$||z_i - z_j|| = dist_{ij}$.

显然,MDS 算法的目标就是从已知条件–原始空间距离矩阵 $\mathbf{D} \in \mathbb{R}^{m\times m}$, 得到 $\mathbf{Z} \in \mathbb{R}^{d'\times m}, d' \leq d$. 即找到两者间的关系.

从已知条件出发:

$$||z_i - z_j|| = dist_{ij} \tag{1}$$

将方程两边平方:

$$dist_{ij}^2 = ||z_i||^2 + ||z_j||^2 - 2z_i^T z_j \tag{2}$$

由此形式可知, 为便于计算, 设一矩阵 $\mathbf{B} = \mathbf{Z}^T\mathbf{Z} \in \mathbb{R}^{m\times m}$,**B** 为降维后样本的内积矩阵, 易知, 对于 **B** 中第 i 行 j 列元素 b_{ij} 有:

$$b_{ij} = z_i^T z_j \tag{3}$$

带入上述方程有:

$$dist_{ij}^2 = b_{ii} + b_{jj} - 2b_{ij} \tag{4}$$

因此问题变成找矩阵 **D** 和矩阵 **B** 之间的关系.

为便于讨论, 令降维后的样本 **Z** 被中心化, 即 $\sum_{i=1}^{m} z_i = 0$. 显然, 矩阵 **B** 的行与列之和均为零, 即 $\sum_{i=1}^{m} b_{ij} = \sum_{j=1}^{m} b_{ij} = 0$.
于是我们得到了一个新的方程组:

$$\begin{cases} dist_{ij}^2 = b_{ii} + b_{jj} - 2b_{ij} \\ \sum_{i=1}^{m} b_{ij} = \sum_{j=1}^{m} b_{ij} = 0 \end{cases} \tag{5}$$

解得:

$$\begin{aligned} \sum_{i=1}^{m} dist_{ij}^2 &= \sum_{i=1}^{m} b_{ii} + \sum_{i=1}^{m} b_{jj} - 2\sum_{i=1}^{m} b_{ij} \\ &\left(\sum_{i=1}^{m} b_{ii} = tr(\mathbf{B}), \sum_{i=1}^{m} b_{ij} = 0\right) \\ &= tr(\mathbf{B}) + mb_{jj} \end{aligned} \tag{6}$$

同理可得:

$$\sum_{j=1}^{m} dist_{ij}^2 = tr(\mathbf{B}) + mb_{ii} \tag{7}$$

$$\begin{aligned} \sum_{i=1}^{m}\sum_{j=1}^{m} dist_{ij}^2 &= \sum_{i=1}^{m} \left(tr(\mathbf{B}) + mb_{ii}\right) \\ &= 2mtr(\mathbf{B}) \end{aligned} \tag{8}$$

其中 tr(·) 表示矩阵的迹,$tr(\mathbf{B}) = \sum_{i=1}^{m} ||z_i||^2$.
由公式 (4),(6) (8) 可知:

$$\begin{cases} b_{ij} = \frac{1}{2}\left(b_{ii} + b_{jj} - dist_{ij}^2\right) \\ b_{ii} = \frac{1}{m}\left(\sum_{j=1}^{m} dist_{ij}^2 - tr(\mathbf{B})\right) \\ b_{jj} = \frac{1}{m}\left(\sum_{i=1}^{m} dist_{ij}^2 - tr(\mathbf{B})\right) \\ tr(\mathbf{B}) = \frac{1}{2m}\left(\sum_{i=1}^{m}\sum_{j=1}^{m} dist_{ij}^2\right) \end{cases}$$

解得:

$$b_{ij} = \frac{1}{2}\left(\frac{1}{m}\sum_{j=1}^{m} dist_{ij}^2 + \frac{1}{m}\sum_{i=1}^{m} dist_{ij}^2 - \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} dist_{ij}^2 - dist_{ij}^2\right)$$

为便于表示,令

$$dist_{i\cdot}^2 = \frac{1}{m}\sum_{j=1}^{m} dist_{ij}^2 \tag{9}$$

$$dist_{\cdot j}^2 = \frac{1}{m}\sum_{i=1}^{m} dist_{ij}^2 \tag{10}$$

$$dist_{\cdot\cdot}^2 = \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} dist_{ij}^2 \tag{11}$$

由此即可通过降维前后保持不变的距离矩阵 **D** 求内积矩阵 **B**:

$$b_{ij} = \frac{1}{2}\left(dist_{i\cdot}^2 + dist_{\cdot j}^2 - dist_{\cdot\cdot}^2 - dist_{ij}^2\right) \tag{12}$$

至此, 在难以直接从矩阵 **D** 中求得矩阵 **Z** 的情况下, 引入了矩阵 $\mathbf{B} = \mathbf{Z}^T\mathbf{Z} \in \mathbb{R}^{m\times m}$ 使得其与矩阵 **D** 具体的联系起来. 更具体来讲, 一个距离矩阵 **D** 与一个内积矩阵 **B** 是一一对应的, 正如本文开头所讲的 MDS 算法的目标那样, 在已知距离矩阵 **D** 的情况下, 就能得到一个确切的内积矩阵 **B**, 而任何满足 $\mathbf{Z}^T\mathbf{Z} = \mathbf{B}$ 的样本矩阵都是原始数据矩阵 **X** 多维缩放的结果. 显然 $\mathbf{B} = \mathbf{X}^T\mathbf{X}$.

对于 d' 维样本矩阵 **Z**:

$$\mathbf{Z} = \begin{bmatrix} z_{11} & z_{21} & \dots & z_{m1} \\ z_{12} & z_{22} & \dots & z_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ z_{1d'} & z_{2d'} & \dots & z_{md'} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} z_{11} & z_{12} & \dots & z_{1d'} \\ z_{21} & z_{22} & \dots & z_{2d'} \\ \vdots & \vdots & \ddots & \vdots \\ z_{m1} & z_{m2} & \dots & z_{md'} \end{bmatrix} \times \begin{bmatrix} z_{11} & z_{21} & \dots & z_{m1} \\ z_{12} & z_{22} & \dots & z_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ z_{1d'} & z_{2d'} & \dots & z_{md'} \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{j=1}^{d'} z_{1j}^2 & \sum_{j=1}^{d'} z_{1j}z_{2j} & \dots & \sum_{j=1}^{d'} z_{1j}z_{mj} \\ \sum_{j=1}^{d'} z_{2j}z_{1j} & \sum_{j=1}^{d'} z_{2j}^2 & \dots & \sum_{j=1}^{d'} z_{2j}z_{mj} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{j=1}^{d'} z_{mj}z_{1j} & \sum_{j=1}^{d'} z_{mj}z_{2j} & \dots & \sum_{j=1}^{d'} z_{mj}^2 \end{bmatrix}$$

从矩阵 **B** 中重新提取出矩阵 **Z**:

尽管上述所说 $\mathbf{B}=\mathbf{Z}^T\mathbf{Z}$, 实际上在 $d'<d$ 时很难实现严格相等,(理论上 $d-d'$ 越大, 误差越大, 因为使用的特征值越少) 所以无论出于理论要求还是实际需求, 我们只需让降维后的距离与原始距离尽可能接近而不必严格相等.

特征值分解:

$$\mathbf{B}=\mathbf{V}\Lambda\mathbf{V}^T$$

其中 $\Lambda=diag(\lambda_1,\lambda_2,\ldots,\lambda_d)$ 为特征值构成的对角矩阵,$\mathbf{V}$ 为特征向量矩阵.

分解得到的矩阵 Λ 是一个对角阵, 里面的特征值是由大到小排列的, 这些特征值所对应的特征向量就是描述这个矩阵变化方向. 我们通过特征值分解得到的前 d' 个特征向量, 那么就对应了这个矩阵最主要的 d' 个变化方向。我们利用这前 d' 个变化方向, 就可以近似这个矩阵 (变换)。

基于此, 求得的矩阵 $\mathbf{Z}$ 即提取矩阵 Λ 中前 d' 个最大的特征值构成新的对角矩阵 $\tilde{\Lambda}$:

$$\begin{aligned}\mathbf{B}&=\mathbf{V}\Lambda\mathbf{V}^T\\&=\mathbf{V}\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}\mathbf{V}^T\\&=(\Lambda^{\frac{1}{2}}\mathbf{V}^T)^T\Lambda^{\frac{1}{2}}\mathbf{V}^T\\&=\mathbf{Z}^T\mathbf{Z}\end{aligned}\tag{13}$$

则 $\mathbf{Z}$ 表示为:

$$\mathbf{Z}=\tilde{\Lambda}^{\frac{1}{2}}\tilde{\mathbf{V}}^T\in\mathbb{R}^{d'\times m}\tag{14}$$

其中 $\tilde{\mathbf{V}}$ 为 $\tilde{\Lambda}$ 对应的特征向量矩阵.

矩阵每列是一个样本的低维坐标.

3 PCA 算法基本原理和证明

输入:

样本集 $D = x_1, x_2, \ldots, x_m$;

低维空间维数 d';

过程:

1. 对所有样本进行中心化:$x_i \leftarrow x_i - \frac{1}{m}\sum_{i=1}^{m} x_i$;
2. 计算样本的协方差矩阵 $\mathbf{X}\mathbf{X}^T$;
3. 对协方差矩阵 $\mathbf{X}\mathbf{X}^T$ 做特征值分解;
4. 取最大的 d' 个特征值所对应的特征向量 $w_1, w_2, \ldots, w_{d'}$;

输出:

投影矩阵 $\mathbf{W}^* = (w_1, w_2, \ldots, w_{d'})$;

一个三维物体, 选取一个面进行投影, 得到的就是一个二维平面.PCA 算法就是一个投影算法, 将一个高维空间投影到一个低维空间当中.

PCA 算法的目标就是找到这样一个超平面, 能够让这个投影保留更大的特征, 对所有样本进行恰当的表达.

那么这个超平面将满足这样的性质:

- 最近重构性: 样本点到这个超平面的距离都足够近;
- 最大可分性: 样本点在这个超平面上的投影能尽可能分开.

实际上, 基于这两种情况将会得到 PCA 的等价推导.

从最近重构性来推导:

对样本进行中心化, 即令 $\sum_i x_i = 0$, 在 d 维空间中我们取一组新的基向量 $\mathbf{W} = \{w_1, w_2, \ldots, w_d\}$, 其中 w_i 是标准正交基向量, 实际上这组基向量即 $\mathbf{W}$ 相当于坐标系的旋转变换, 而这个新的'坐标系'即 $\mathbf{W}$ 是不确定的,

PCA 假设原空间存在一个线性变换 $\mathbf{W}$, 从上面取得 d' 个最大特征值的特征向量方向 (此时投影矩阵为 $\mathbf{W}^* = (w_1, w_2, \ldots, w_{d'})$, 接下来的 $\mathbf{W}$ 皆指矩阵 $\mathbf{W}^*$), 让原空间的元素投影到这个 d' 空间, 并让元素之间距离差之和最小.

设样本点 x_i 在 d' 维空间的投影为 $z_i = (z_{i1}; z_{i2}; \ldots; z_{id'})$, 因为 w_i 是正交基向量, 所以:

$$
\begin{aligned}
z_{ij} &= w_j^T x_i \\
z_i &= \mathbf{W}^T x_i \qquad (15)
\end{aligned}
$$

此处 $\mathbf{W}=(w_1,w_2,\ldots,w_{d'})$

显然 z_i 是降维后的样本在 d' 维空间的坐标表示, 即基于这 d' 个基向量的空间表示, 无法直接与原样本进行距离计算.

因此我们需要基于 z_i 重构 $\hat{x_i}$:

$$\begin{aligned}\hat{x_i} &= \sum_{j=1}^{d'} z_{ij} w_j \\ &= \mathbf{W} z_i \\ &= \mathbf{W}\mathbf{W}^T x_i\end{aligned} \tag{16}$$

因此就得到一个优化问题:

$$\begin{aligned}\underset{\mathbf{W}}{min} \quad & \sum_{i=1}^{m} \left|\left|\mathbf{W}\mathbf{W}^T x_i - x_i\right|\right|_2^2 \\ s.t. \quad & \mathbf{W}^T\mathbf{W}=\mathbf{I}\end{aligned} \tag{17}$$

已知 $\mathbf{W}^T\mathbf{W}=\mathbf{I}$, $z_i=\mathbf{W}^T x_i$, 则:

$$\begin{aligned}\sum_{i=1}^{m} \left|\left|\mathbf{W}\mathbf{W}^T x_i - x_i\right|\right|_2^2 &= \sum_{i+1}^{m} \left(\mathbf{W}\mathbf{W}^T x_i - x_i\right)^T \left(\mathbf{W}\mathbf{W}^T x_i - x_i\right) \\ &= \sum_{i=1}^{m} \left(x_i^T\mathbf{W}\mathbf{W}^T\mathbf{W}\mathbf{W}^T x_i - x_i^T\mathbf{W}\mathbf{W}^T x_i - x_i^T\mathbf{W}\mathbf{W}^T x_i + x_i^T x_i\right) \\ &= \sum_{i=1}^{m} \left(-x_i^T\mathbf{W}\mathbf{W}^T x_i + x_i^T x_i\right) \\ &= -\sum_{i=1}^{m} x_i^T\mathbf{W}\mathbf{W}^T x_i + \sum_{i=1}^{m} x_i^T x_i \\ &= -\sum_{i=1}^{m} \left(\mathbf{W}^T x_i\right)^T \mathbf{W}^T x_i + const \\ &= -\sum_{i=1}^{m} tr\left(\mathbf{W}^T x_i x_i^T \mathbf{W}\right) + const \\ &\approx -tr\left(\mathbf{W}^T \left(\sum_{i=1}^{m} x_i x_i^T\right)\mathbf{W}\right)\end{aligned} \tag{18}$$

考虑到 w_j 是标准正交基, $\sum_i x_i x_i^T$ 是协方差矩阵, 有:

$$\begin{aligned}\underset{\mathbf{W}}{min} \quad & -tr\left(\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W}\right) \\ s.t. \quad & \mathbf{W}^T\mathbf{W}=\mathbf{I}\end{aligned} \tag{19}$$

对于带矩阵约束的优化问题, 其优化目标的拉格朗日函数为

$$\begin{aligned} L(\mathbf{W}, \boldsymbol{\theta}) &= -tr(\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W}) + \langle \boldsymbol{\theta}, \mathbf{W}^T\mathbf{W} - \mathbf{I} \rangle \\ &= -tr(\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W}) + \big(\boldsymbol{\theta}^T(\mathbf{W}^T\mathbf{W} - \mathbf{I})\big) \end{aligned} \tag{20}$$

其中,$\boldsymbol{\theta} \in \mathbb{R}^{d' \times d'}$ 为拉格朗日乘子矩阵, 其维度恒等于约束条件的维度, 且其中的每个元素均为未知的拉格朗日乘子.

$\langle \boldsymbol{\theta}, \mathbf{W}^T\mathbf{W} - \mathbf{I} \rangle$ 为矩阵的内积.

若此时仅考虑约束 $w_i^T w_i = 1 (i = 1, 2, \ldots, d')$, 则拉格朗日乘子矩阵 $\boldsymbol{\theta}$ 为对角矩阵, 令新的拉格朗日乘子矩阵为 $\mathbf{\Lambda} = diag(\lambda_1, \lambda_2, \ldots, \lambda_{d'}) \in \mathbb{R}^{d' \times d'}$, 则新的拉格朗日函数为

$$L(\mathbf{W}, \mathbf{\Lambda}) = -tr(\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W}) + \big(\mathbf{\Lambda}^T(\mathbf{W}^T\mathbf{W} - \mathbf{I})\big) \tag{21}$$

对拉格朗日函数关于 $\mathbf{W}$ 求导可得

$$\begin{aligned} \frac{\partial L(\mathbf{W}, \mathbf{\Lambda})}{\partial \mathbf{W}} &= \frac{\partial}{\partial \mathbf{W}} \left[-tr(\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W}) + tr\big(\mathbf{\Lambda}^T(\mathbf{W}^T\mathbf{W} - \mathbf{I})\big)\right] \\ &= -\frac{\partial}{\partial \mathbf{W}} tr(\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W}) + \frac{\partial}{\partial \mathbf{W}} tr\big(\mathbf{\Lambda}^T(\mathbf{W}^T\mathbf{W} - \mathbf{I})\big) \\ &= -2\mathbf{X}\mathbf{X}^T\mathbf{W} + 2\mathbf{W}\mathbf{\Lambda} \end{aligned} \tag{22}$$

令 $\frac{\partial L(\mathbf{W}, \mathbf{\Lambda})}{\partial \mathbf{W}} = 0$ 可得

$$\begin{aligned} -2\mathbf{X}\mathbf{X}^T\mathbf{W} + 2\mathbf{W}\mathbf{\Lambda} &= 0, \\ \mathbf{X}\mathbf{X}^T\mathbf{W} &= \mathbf{W}\mathbf{\Lambda} \end{aligned} \tag{23}$$

其中,λ_i 和 w_i 分别表示矩阵 $\mathbf{X}\mathbf{X}^T$ 的特征值和单位特征向量.

已知,$\mathbf{X}\mathbf{X}^T$ 是一个实对称矩阵, 实对称矩阵的不同特征值所对应的特征向量之间相互正交, 所以通过上式求得的 w_i 可以同时满足约束 $w_i^T w_i = 1$ 和 $w_i^T w_j = 0 (i \neq j)$.

将其带回目标函数中得

$$\begin{aligned} \underset{\mathbf{W}}{min} \quad - tr\left(\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W}\right) &= \underset{\mathbf{W}}{max} \quad tr\left(\mathbf{W}^T\mathbf{W}\mathbf{\Lambda}\right) \\ &= \underset{\mathbf{W}}{max} \quad tr(\mathbf{\Lambda}) \\ &= \underset{\mathbf{W}}{max} \quad \sum_{i=1}^{d'} \lambda_i \end{aligned} \tag{24}$$

所以选择最大的 d' 个特征值就能使函数达到最优值.

取这 d' 个特征值对应的特征向量 w_i 即可得到投影矩阵 $\mathbf{W} = (w_1, w_2, \ldots, w_{d'})$, 这就是主成分分析的解.

对于低维空间维数 d' 通常是由用户事先指定, 或通过在 d' 值不同的低维空间中对 k 近邻分类器 (或其他开销较小的学习器) 进行交叉验证来选取较好的 d' 值.

对 PCA, 还可从重构的角度设置一个重构阈值, 例如 t=95%, 然后选取使下式成立的最小 d' 值:

$$\frac{\sum_{i=1}^{d'} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \geq t. \tag{25}$$